DOI: 10.54254/2755-2721/63/20240988

Wind speed prediction

Alvin Xianghan Li

Nanjing Foreign Language School, Xuanwu District, Nanjing City, Jiangsu Province, China, 210000

Xianghanli828@gmail.com

Abstract. With the help of wind farms, wind energy is a vital renewable energy source that contributes significantly to the world's energy balance. The lifespan and maintenance costs of wind turbines will be reduced with an accurate wind speed prediction. On the other hand, wind speed is highly volatile and unpredictable. Thus, it is essential to do research into creating complex models and algorithms for precise wind speed prediction. So far, some of the most promising models include Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Autoregressive Moving Average (ARMA). Python, as an advanced and versatile programming language, is exceptionally suited for scripting the algorithms of these sophisticated models. This paper will use the data from Austin Texas and apply a Support Vector Machine (SVM) for wind speed prediction involves several stages, including data collection, data preprocessing, model selection, model training, parameter optimization, model validation, and prediction. Wind energy resource optimisation, maintenance cost reduction, and total wind farm efficiency can all be significantly improved by incorporating these models into predictive analytics and continuously improving them against changing data.

Keywords: Wind energy, speed prediction, Support Vector Machine

1. Introduction

Especially in complicated and intense weather scenarios, current predictive methods for wind speed estimation are unable to adequately forecast wind conditions. This suggests that to increase the accuracy of wind speed predictions, it is imperative to further improve already-existing algorithms or create new models. However, this work must employ a highly complex model and compare the results to one another to obtain an accurate result [1].

Although there has been notable progress in the development of wind speed prediction models, these models still lack the necessary precision, particularly when complicated and harsh weather conditions are present. The necessity of improving the current models is highlighted by their unreliability in the face of erratic weather patterns. The unreliability of current models in the face of unpredictable weather patterns underscores the urgency for refining the existing algorithms or fabricating novel models, which can enhance the accuracy of wind speed predictions. The task of improving these predictive models, however, is challenging. A more complex and intricate model might be required to encapsulate the multiplicity of factors that interact to generate variable wind speeds [2]. This would entail the integration of multiple variables, possibly leading to a high-dimensional, sophisticated model. It is important, nonetheless, to note that complexity does not always correlate with accuracy. Increased complexity could, inversely, lead to overfitting or increased computational demands. Therefore, the development and application of more complex models should be complemented by a robust comparative evaluation [3]. This would refer to comparing the newly developed complex models against their simpler counterparts or even across multiple complex models. The outputs of these evaluations should then be analyzed side by side to decide which model presents the best blend of precision, efficiency, and reliability [4]. The journey towards fabricating a accurate wind speed prediction model is still ongoing. While reasonable headways have been achieved, the quest to uncover an ideal predictive model that captures the capricious nature of wind speeds continues, becoming increasingly critical in our current era of reliance on renewable energy [5]. Pioneering breakthroughs in this field would not only bolster our meteorological prowess but also offer tangible benefits for various sectors, such as in wind energy harvesting, where precise wind speed predictions are key to optimizing output and efficiency [6].

2. Support Vector Machine

This paper starts by gathering historical records of wind speed. Data may include wind speed, date, time, temperature, humidity, atmospheric pressure, and any other factors people believe might affect the wind speed [7]. Clean and format people's data to meet the needs of the model. Preprocessing might involve deleting or filling missing data, transforming categorical data into numerical form, and normalizing or standardizing numerical data. Additionally, the data perhaps needs to be divided into a feature set and labels [8]. In this case, wind speed would be the label, and all other factors would form the feature set. This paper uses a Support Vector Machine (SVM) classifier, which is a supervised learning model suitable for classification or regression predictions. Create an SVM object and train it with the feature set and labels.

This paper utilizes techniques such as grid search or cross-validation to optimize the model parameters. The aim is to find the best set of parameters that maximize the predictive performance. Model Validation: Evaluate performance by comparing the model against a test set [9]. A model that performs well on unseen data, indicating good generalization, can be used for practical wind speed prediction. Prediction of New Data: When people are satisfied with the model's performance, it can be applied to new data (like real-time weather data) to predict future wind speeds. Remember, the creation and optimization of predictive models take time and may require several iterations and refinements. In practice, the model should be trained regularly to maintain its predictive capabilities [10].

3. Random Forest

Accurate categorization has significantly improved when an ensemble of trees is grown and allowed to vote for the class that they find most appealing. Random vectors that control the growth of every tree in the ensemble are created to expand these ensembles. An early example is bagging, in which each tree is grown by randomly selecting (without replacement) from the training set of examples.

Another illustration is random split selection, in which the split is chosen at random from the K best splits at each node. By introducing randomness into the original training set's outputs, Breiman creates new training sets [11]. An alternative method would be to choose the training set at random using weights assigned to the training set's examples. His "random subspace" method, which chooses a subset of characteristics at random to utilise in growing each tree, has been the subject of multiple articles.

An algorithm for machine learning based on ensemble trees is called the Random Forest Classifier. A collection of decision trees drawn at random from a subset of the training set makes up the random forest classifier. To determine the test object's final class, it adds up the votes from various decision trees [12].

Ensemble algorithms incorporate many object classification techniques, either of the same or distinct types. As an illustration, consider predicting using naive Bayes, SVM, and decision trees, and then voting on the test object's final class consideration.

DOI: 10.54254/2755-2721/63/20240988

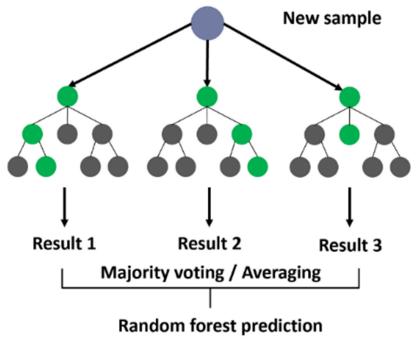

Figure 1. Random forest prediction [13]

As shown in Figure 1, random Forest works by creating a multitude of decision trees during the training phase. The output for classification is determined by the mode of the classes (the class selected by the majority of trees) and for regression by the mean or average prediction of the individual trees. Here's a step-by-step breakdown of its operation:

Bootstrap sampling: For each decision tree, a sample of the training data is taken with replacement, known as bootstrap sampling.

Tree creation: On these bootstrapped samples, decision trees are grown. Instead of using all features to split at each node, Random Forest randomly selects a subset of features for each tree at each split [14].

Prediction: For classification, each decision tree in the forest outputs a class prediction, and the most popular class among all trees is considered the final prediction. In regression, it averages the predictions from all the trees to determine the final prediction.

Ensemble: The ensemble of decision trees reduces variance and avoids overfitting, which is a common problem with individual decision trees [15].

Random Forest can be effectively utilized for forecasting wind speed, which is a crucial parameter in meteorology, safety, and energy production, particularly for wind farms. Here's how Random Forest can be used for the task:

Data Collection and Preprocessing: Gather historical wind speed data along with other relevant features such as temperature, pressure, humidity, date/time, etc [16].

Feature Selection: Use Random Forest for feature importance to select the most relevant predictors for wind speed.

Model Training: Train the Random Forest model using historical data. During this phase, different hyperparameters can be tuned, such as the number of trees, depth of trees, or the number of features considered for splitting at each node [17].

Model Validation: Validate the model using a separate validation dataset to check its predictive performance and make necessary adjustments.

Wind Speed Forecasting: Apply the trained model to new data to predict future wind speeds.

Random forest is an ensemble machine learning algorithm that is well-suited for a variety of prediction tasks, including wind speed prediction. Here are several advantages of using random forest for this application:

Handling Non-linearity: Wind speed data is often influenced by multiple factors leading to non-linear relationships. Random forests can capture these complex interactions between variables without the need for transformation of variables, making it a robust predictive model for wind speed.

Reduced Overfitting: Through the creation of numerous decision trees and averaging their results, random forests naturally reduce the risk of overfitting compared to individual decision trees. This is particularly useful in wind speed prediction, where the data can be noisy and prone to overfit.

Feature Importance: The random forest algorithm can rank the importance of different weather parameters affecting wind speed. This feature importance score provides insights into which factors contribute most to the variability in wind speed, aiding in the understanding and improvement of the models.

Flexibility with Data Types: Random forests can handle both categorical and numerical data. This is useful in the context of wind speed prediction, where inputs might include various data types such as wind direction (categorical) and atmospheric pressure (continuous).

Minimal Need for Data Preprocessing: Random forests do not require scaling of data. Many algorithms, like SVM or neural networks, require the normalization of data, but random forests can work well with variables of varying scales without the need for normalization.

4. Conclusion

In conclusion, the pursuit of accurate wind speed prediction models is an essential endeavor within the realm of renewable energy. Despite the plethora of methods available, such as SVM and random forests, perfecting these models remains a challenge due to the complex and volatile nature of wind patterns. Through the thorough methodology detailed in this paper—encompassing robust data collection, meticulous preprocessing, and the utilization of advanced machine learning techniques—significant strides can be made towards more precise wind speed forecasts. Support Vector Machine (SVM) and Random Forest (RF) approaches have been highlighted, each with distinct advantages. SVM's potent capability for regression in high-dimensional spaces complements Random Forest's robustness in handling non-linearity and reducing overfitting. By integrating these models into predictive analytics, and consistently refining them against evolving data, a substantial impact can be made on optimizing wind energy resources, mitigating maintenance costs, and enhancing the overall efficiency of wind farms. The future of wind speed prediction looks promising as this paper leverages the computational power and sophistication of machine learning algorithms to harness the true potential of wind as a sustainable energy resource.

References

[1] Theodoridis, Sergios; and Koutroumbas, Konstantinos; "Pattern Recognition", 4th Edition, Academic Press, 2009, ISBN 978-1-59749-272-0
[2] Cristianini, Nello; and Shawe-Taylor, John; An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, 2000. ISBN 0-521-78019-5 (SVM Book)
[3] Huang, Te-Ming; Kecman, Vojislav; and Kopriva, Ivica (2006); Kernel Based Algorithms for Mining Huge Data Sets, in Supervised, Semi-supervised, and Unsupervised Learning, Springer-Verlag, Berlin, Heidelberg, 260 pp.96 illus., Hardcover, ISBN 3-540-31681-7
[4] Kecman, Vojislav; Learning and Soft Computing— Support Vector Machines, Neural Networks, Fuzzy Logic Systems, The MIT Press, Cambridge, MA, 2001.
[5] Schölkopf, Bernhard; and Smola, Alexander J.; Learning with Kernels, MIT Press, Cambridge, MA, 2002. ISBN 0-262-19475-9
[6] Schölkopf, Bernhard; Burges, Christopher J. C.; and Smola, Alexander J. (editors); Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, MA, 1999. ISBN 0-262-19416-3.
[7] Shawe-Taylor, John; and Cristianini, Nello; Kernel Methods for Pattern Analysis, Cambridge University Press, 2004. ISBN 0-521-81397-2 (Kernel Methods Book)
[8] Steinwart, Ingo; and Christmann, Andreas; Support Vector Machines, Springer-Verlag, New York, 2008. ISBN 978-0-387-77241-7 (SVM Book)
[9] Tan, Peter Jing; and Dowe, David L. (2004); MML Inference of Oblique Decision Trees, Lecture Notes in Artificial Intelligence (LNAI) 3339, Springer-Verlag, pp. 1082–1088. (This paper uses minimum message length (MML) and incorporates probabilistic support vector machines in the leaves of decision trees.)
[10] Vapnik, Vladimir N.; The Nature of Statistical Learning Theory, Springer-Verlag, 1995. ISBN 0-387-98780-0

DOI: 10.54254/2755-2721/63/20240988

[11] Vapnik, Vladimir N.; and Kotz, Samuel; Estimation of Dependences Based on Empirical Data, Springer, 2006. ISBN 0-387-30865-2, 510 pages [this is a reprint of Vapnik's early book describing philosophy behind SVM approach. The 2006 Appendix describes recent development].
[12] Fradkin, Dmitriy; and Muchnik, Ilya; Support Vector Machines for Classification in Abello, J.; and Carmode, G. (Eds); Discrete Methods in Epidemiology, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, volume 70, pp.13–20, 2006. Succinctly describes theoretical ideas behind SVM.
[13] Bennett, Kristin P.; and Campbell, Colin; Support Vector Machines: Hype or Hallelujah? SIGKDD Explorations, 2, 2, 2000, 1–13. Excellent introduction to SVMs with helpful figures.
[14] Ivanciuc, Ovidiu; Applications of Support Vector Machines in Chemistry, in Reviews in Computational Chemistry, Volume 23, 2007, pp.291–400.
[15] Catanzaro, Bryan; Sundaram, Narayanan; and Keutzer, Kurt; Fast Support Vector Machine Training and Classification on Graphics Processors, International Conference on Machine Learning, 2008
[16] Campbell, Colin; and Ying, Yiming; Learning with Support Vector Machines, 2011, Morgan and Claypool. ISBN 978-1-60845-616-1.
[17] Ben-Hur, Asa, Horn, David, Siegelmann, Hava, and Vapnik, Vladimir; "Support vector clustering" (2001) Journal of Machine Learning Research, 2: 125–137.